

(Часть 2) Практические задачи

Для данных задач в ваших гитах первого этапа выделите отдельную папку (package)

Это необходимо, чтобы не смешивать с заданиями из первой части задач.

1. Вычислить ряд Фибоначчи.

Ряд Фибоначчи - последовательность чисел, в которой любое число, кроме первого и второго, является суммой двух предыдущих: 0, 1, 1, 2, 3, 5, 8, 13, 21...

Количество чисел в ряду Фибоначчи ввести с консоли.

2. Прочитать с экрана консоли произвольный текст, содержащий не только буквы, но цифры и другие символы. Удалить из введенного текста все символы, кроме пробелов, не являющиеся буквами.

3. Замена символов в словах по позиции (шаги программы):

- Прочитать с экрана консоли текст из нескольких слов различной длины (разделитель между словами один пробел);
- Прочитать с экрана консоли номер k-позиции символа в словах и новый символ для замены;
- В каждом слове введенного текста заменить k-ю букву на заданный символ. Если k больше длины слова, слово пропускать.

4. Прочитать с экрана консоли текст из любого количество слов. Найти и вывести на экран все слова максимальной и все слова минимальной длины.

5. Прочитать с экрана консоли текст, состоящий из слов. Преобразовать предложение в следующую модель классов.

Создать классы (диаграмма 1):

- Класс "Sentence"; //предложение
- Класс "Word"; //слово

Класс "Sentence" в качестве поля класса (переменной экземпляра) должен содержать массив классов "Word"; конструктор в качестве аргумента должен принимать массив String[] слов исходного предложения; данный массив надо преобразовать в массив "Word" и присвоить полю класса.

- Снабдить класс "Sentence" public методом, который возвращает количество слов в предложении.

- Снабдить класс "Sentence" public методом, который возвращает экземпляр класса "Word" из массива слов аргумента класса по его позиции в предложении.

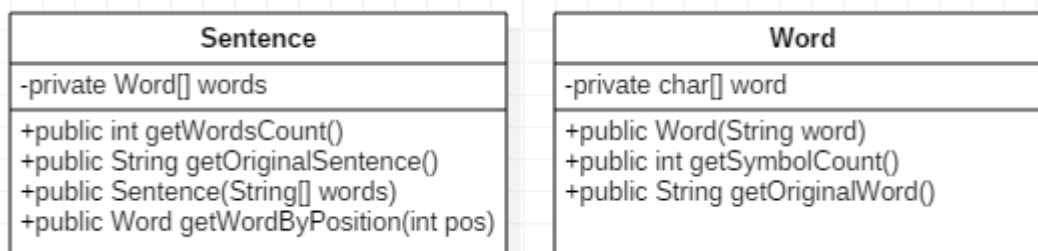
- Снабдить класс "Sentence" public методом, который воссоздает исходное предложение из массива слов Word[].

Класс "Word" в качестве поля класса (переменной экземпляра) должен содержать массив символов слова char[]; конструктор в качестве аргумента должен принимать слово String исходного предложения; слово надо преобразовать в массив char[] и присвоить полю класса.

- Снабдить класс "Word" public методом, который возвращает количество символов в слове.

- Снабдить класс "Word" public методом, который воссоздает слово в виде String из массива char[].

Диаграмма 1. Классы программы.



После создания модели вывести на консоль сообщение:
Модель создана.

P.S.

Данную задачу мы будем развивать и модифицировать двумя спринтами.